CSE142 Sample Midterm Key
Autumn 2019

1.

Expression	Value
`3 * 4 + 5 * 6`	42
`23 % 5 - 17 % (16 % 10)`	-2
`"1" + 2 + 3 * 4 + (5 + 6)`	"121211"
`1.5 * 2 + 20 / 3 / 4.0 + 6 / 4`	5.5
`345 / 10 / 3 + 10 / (5 / 2.0)`	15.0

2. The program produces the following output:

```
semi missing a brace and a 42
semi missing a 42 and a 8
brace missing a literal and a semi
84 missing a 1 and a cse
```

3.

Method Call	Output Produced
`ifElseMystery(2, 7);`	11 7
`ifElseMystery(6, 6);`	6 16
`ifElseMystery(4, -1);`	3 0
`ifElseMystery(11, 10);`	11 21
`ifElseMystery(-10, 7);`	0 17
`ifElseMystery(100, 5);`	99 6

4.

Method Call	Output Produced
`mystery(0);`	1 1
`mystery(7);`	2 12
`mystery(32);`	3 123
`mystery(256);`	4 1234

5.

	num < 0	first < second	num >= second
Point A	sometimes	never	always
Point B	never	sometimes	sometimes
Point C	never	always	never
Point D	never	sometimes	sometimes
Point E	always	sometimes	sometimes

6. Two possible solutions appear below.

```
    public static void spinWheel(Random r, int n) {
        int spin = r.nextInt(5) * 10 + 20;
        System.out.print("spins: " + spin);
        int count = 0;
        if (spin == 20) {
            count++;
        }
        int totalSpins = 1;
        while (count < n) {
            spin = r.nextInt(5) * 10 + 20;
            totalSpins++;
            System.out.print(", " + spin);
            if (spin == 20) {
                count++;
            } else {
                count = 0;
            }
        }
        System.out.println();
        System.out.println(n + " in a row after " + totalSpins + " spins");
    }

    // solution that prints solves the fencepost by printing 20 after the
    // loop:
    public static void spinWheel(Random r, int n) {
        System.out.print("spins: ");
        int count = 0;
        int totalSpins = 0;
        while (count < n) {
            int spin = r.nextInt(5) * 10 + 20;
            totalSpins++;
            if (spin == 20) {
                count++;
            } else {
                count = 0;
            }
            if (count < n) {
                System.out.print(spin + ", ");
            }
        }
        System.out.println(20);
        System.out.println(n + " in a row after " + totalSpins + " spins");
    }
```

7. One possible solution appears below.

```
public static boolean balanceCheckbook(Scanner console) {
    System.out.print("initial balance? ");
    double balance = console.nextDouble();
    System.out.print("how many transactions? ");
    int count = console.nextInt();
    double min = balance;
    for (int i = 1; i <= count; i++) {
        System.out.print(i + "/" + count + " amount? ");
        double amount = console.nextDouble();
        balance = balance + amount;
        System.out.println("new balance = $" + balance);
        if (balance < min) {
            min = balance;
        }
    }
    System.out.println("minimum balance = $" + min);
    return (min < 0);
}
```

8. Three possible solutions appear below.

```
    public static String hashTag(String s) {
        String result = "";
        boolean firstLetter = true;
        for (int i = 0; i < s.length(); i++) {
            if (s.charAt(i) == ' ') {
                firstLetter = true;
            } else {
                if (firstLetter) {
                    result += Character.toUpperCase(s.charAt(i));
                    firstLetter = false;
                } else {
                    result += Character.toLowerCase(s.charAt(i));
                }
            }
        }
        return "#" + result;
    }

    public static String hashTag(String s) {
        String result = "";
        if (s.length() > 0) {
            if (s.charAt(0) != ' ') {
                result += Character.toUpperCase(s.charAt(0));
            }
        }
        for (int i = 1; i < s.length(); i++) {
            if (s.charAt(i - 1) == ' ' && s.charAt(i) != ' ') {
                result += Character.toUpperCase(s.charAt(i));
            } else if (s.charAt(i) != ' ') {
                result += Character.toLowerCase(s.charAt(i));
            }
        }
        return "#" + result;
    }

    public static String hashTag(String s) {
        String result = "";
        int space = s.indexOf(" ");
        while (s.length() > 0) {
            while (space == 0) { // to trim spaces
                s = s.substring(1);
                space = s.indexOf(" ");
            }
            if (s.length() > 0) {
                String word = s;
                if (space != -1) {
                    word = s.substring(0, space);
                }
                char firstLetter = Character.toUpperCase(word.charAt(0));
                result += firstLetter + word.toLowerCase().substring(1);
                s = s.substring(word.length());
                space = s.indexOf(" ");
            }
        }
        return "#" + result;
    }
```